DOCUMENT RESUME

ED 259 720 IR 011 772

AUTHOR Nugent, Gwen; And Others
TITLE Instructional Development for Videotex: Flowcharts and Scripting. Report Number Three of the Electronic Text Report Series.
INSTITUTION Nebraska Univ., Lincoln. Nebraska Educational Television Network.
SPONS AGENCY Electronic Text Consortium.
PUB DATE 84
NOTE 28p.; For related documents, see IR 011 729 and IR 011 773-775, IR 011 802-803, and ED 255 181. The program on Electronic Text for Higher Education is supported by the Annenberg/CPB Project.
PUB TYPE Guides - Non-Classroom Use (055)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS *Computer Assisted Instruction; Computers; *Flow Charts; Higher Education; *Instructional Design; *Instructional Development; Programing; *Scripts; Teamwork; *Videotex
IDENTIFIERS *Electronic Text; Interactive Systems

ABSTRACT

Designed to assist in the construction of flowcharts and scripts for videotex production, this document presents a detailed look at the instructional design and production of educational videotex. Included are: a flowchart of the videotex development process; hints for effective flowcharting and script development, including frame development; evaluation activities; the definition of computer symbols used within the flowchart; and the suggested structure of personnel teams involved in the project. Sample flowcharts illustrate interactive strategy, showing both forward and backward branching, and branching based on the value of a stored response. Specific suggestions for effective scripts include: non-dense frames, concise and tightly edited text, use of a three-column script form or a script format with grid (samples included). The interdependent relationship of flowchart to script is stressed as a crucial element in the development process. A summary and bibliography complete the report. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

Electronic Text Consortium

Electronic Text for Higher Education

San Diego State University, University of Wisconsin, University of Nebraska, WGBH Boston

ED259720

INSTRUCTIONAL DEVELOPMENT FOR VIDEOTEX: FLOWCHARTS AND SCRIPTING

[Report Number Three]
of the
[Electronic Text Report Series]

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

- This document has been reproduced as received from the person or organization originating it.
- Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

John Witherspoon

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR 011 772

The Program on Electronic Text for Higher Education is supported by
The Annenberg/CPB Project

INSTRUCTIONAL DEVELOPMENT FOR VIDEOTEX: FLOWCHARTS AND SCRIPTING

[Report Number Three]
of the
[Electronic Text Report Series]

Prepared by

Gwen Nugent, P.J. Peters, and Lee Rockwell
Nebraska Educational Television Network
University of Nebraska-Lincoln

for the
Electronic Text Consortium

An Annenberg/CPB Project

The mandate of The Annenberg/CPB Project over a 15-year period is to create innovative, quality college-level materials and to demonstrate unique applications of telecommunications technologies in higher education problems.

1984

IR011772

INSTRUCTIONAL DESIGN FOR VIDEOTEX: FLOWCHARTS AND SCRIPTING

INTRODUCTION

Funded by Cox Cable Communications and the Corporation for Public Broadcasting, the Nebraska Interactive Cable Project began in March, 1982, with the goal of investigating the educational potential of interactive cable. One of the project's tasks was to develop a design and production process for educational uses of videotex. This process was developed through preparation of a series of interactive courses of instruction designed for Cox's two-way interactive cable system, INDAX. Figure 1 graphically depicts the design and production model developed and used by the Nebraska project. Although this paper focuses on the specific steps of flowcharting and scripting, an overview of the total process is presented to provide a better understanding of relationships between these activities and the other components.

OVERVIEW OF VIDEOTEX INSTRUCTIONAL DEVELOPMENT

Developing videotex instruction involves many of the same steps as designing any media product. There should be a preliminary needs analysis to identify exactly what learner needs should be addressed through an instructional product. An audience analysis is also necessary, as well as the development of goals and objectives. One difference, however, is that design models generally include a step known as "select media." Since this paper is concerned only with videotex instruction, this step has been omitted.

In its place, Figure 1 specifies that there should be an analysis of the videotex system. Since options differ with different videotex

Figure 1: Videotex Development Process

systems, the designer cannot ignore the parameters imposed by the hardware configuration. Factors such as limited frame storage, lack of grading capabilities, etc. can restrict the instructional design. Without thorough understanding of the capabilities of videotex systems, there is the risk of designing materials with conventions that cannot be achieved with certain systems. There is a clear interaction between technical capability and instructional and graphic design, and these ramifications must be kept firmly in mind throughout the development process.

Once a videotex system has been selected and there is a clear understanding of what it can and cannot do, planning of instructional strategies can begin. Since videotex is a computer-based system, it can present drill and practice, tutorial and simulation/gaming materials typically associated with computer-assisted instruction. As with any medium, however, the selection of appropriate strategies must take into account the nature of the learner and the content objectives. Other critical factors are the level of interaction required from the user, where feedback should be presented and what form it should take.

When the basic approach and strategies have been determined, flowcharts can be developed. A flowchart serves two basic purposes:

(1) It provides a structure for content development and scripting.

(2) It serves as a guide for preparation of the computer programming.

A general rule is that the more complicated the interactive/branching structure, the more critical the flowchart. For simple sequences the need for a flowchart becomes less important, and

in some cases it may even be omitted in the interest of saving time. If one is not prepared, however, some branching or routing notation must be incorporated into the script so that someone unfamiliar with the content can follow the branching pattern.

Content and flowchart development have been listed together in Figure 1 because, in most cases, the two go hand in hand. A content package for interactive instruction is intrinsically different from that typically prepared for television and film. Content for a film can be linearly prepared with a beginning, middle and end (like a book) and then turned over to a scriptwriter or instructional designer to package for the medium. Content for an interactive program, on the other hand, must be developed for each possible branch and should show the relationship between branches. Although there is generally only one beginning, there may be numerous "middles" and "ends", and considerable effort is often spent in developing "wrong" answer branches. It is this structure which becomes the basic flowchart. The flowchart becomes the roadmap to guide more detailed content development, scripting and, eventually, the computer programming.

Script development follows flowcharting. The script contains information necessary for production of the individual frames. Scripts for interactive videotex programs bear little resemblance to those of television and film. Videotex scripts can be nonlinear and can contain hundreds of sequences whose relationship is only clear from examination of the flowchart. The script sequence is often not the same as the order in which the user will actually view the materials. Scripting requires great attention to detail and meticu-

lous proofing. Scripts can run a thousand pages, and it becomes a true challenge to properly account for all elements and their relationship.

With script completion, the instructional design phase is completed and production can begin. A major production element is frame development. Screens are prepared as independent units and can incorporate visuals, so graphic design is an important component. Frame development requires a skilled artist who is thoroughly familiar with the technical parameters of the system and comfortable with developing graphics using computers instead of pen and ink. For example, the alphamosaic standard in INDAX imposes strong graphic limitations. The designer must be sure that the intended graphics will be compatible with the system.

In addition to frame development, the other production-related function involves computer programming. It is necessary that the branching networks, needed scorekeeping, etc., be entered in a form that the host computer can accept and execute. It should be emphasized that the programming function is not analogous to writing and entering a typical computer program. Computer programs generally include screen display commands in addition to branching and scorekeeping information, but in videotex development, screen design is an independent process.

Once the frames and computer programming have been entered, the instructional programs are complete and may be reviewed and debugged. Debugging "simply" involves checking all the branch points to be sure that they route students to the correct place. If any student responses are to be recorded and stored, the program should also be

checked to see that such functions are executing correctly. Although this step is somewhat analogous to correcting errors in traditional mediated presentations, the debugging process is generally much more extensive and time-consuming. It is not uncommon for debugging to take just as long as, or longer than, developing and entering the original computer program. Since computer programming requires the input of numerous numbers or codes, it is very easy to make a mistake, and it is difficult to spot mistakes once they are entered. Unfortunately, a one-number error can route a student onto a completely wrong branch, where it is impossible for him or her to find the way back to the correct point.

Since videotex production is a time-consuming process, it is important that problems be identified and appropriate revisions made as early as possible. Therefore, evaluation is a part of the development process because needed revisions require recycling through appropriate design-related steps. Evaluation activities can include reviews by students or experts of content packages, flowcharts, scripts, individual frames and fully assembled programs. One likely candidate for preproduction evaluation is the branching structure, since problems there have implications for content, computer programming and frame design. Early evaluations can isolate dead-end branches, closed loops, unclear transitions and confusing structure.

One simple way to test the branching structure is to use program scripts and simulate the branching by listing page numbers where students should be routed beside each choice at a decision point. This scrambled book approach follows the principle of branched programmed

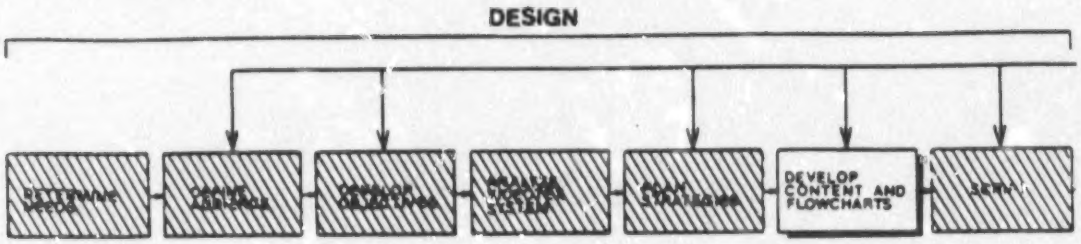
instruction. Students (or experts) can make their choice and proceed to the page indicated. With long scripts this method results in considerable paper shuffling, but it does test the branching structure, as well as providing information on program clarity and content accuracy.

The Team Approach

From this discussion of videotex development, it should be obvious that one person cannot embody all the talents needed to complete a videotex instructional sequence. Therefore, several people must work as a unit. The team approach has long been advocated for the development of instruction utilizing new technologies. As early as Sesame Street days, experts recognized the value of a team approach: "As technologically sophisticated forms of instruction come into increasing prominence, it will be necessary to make increased use of production teams whose members possess a diversity of highly specialized talents" (Palmer, 1974, p. 325).

Videotex teams may differ in complexion depending on the nature of the project and the expertise of team members, but the functions outlined earlier must be accomplished. Teams may include content consultants, instructional designers, scriptwriters, graphic designers, computer programmers and evaluators. Sometimes one person may wear two "hats", and because there are no clear-cut lines between the various functions, it often becomes difficult to determine where one person's responsibility begins and another's ends.

FLOWCHARTING

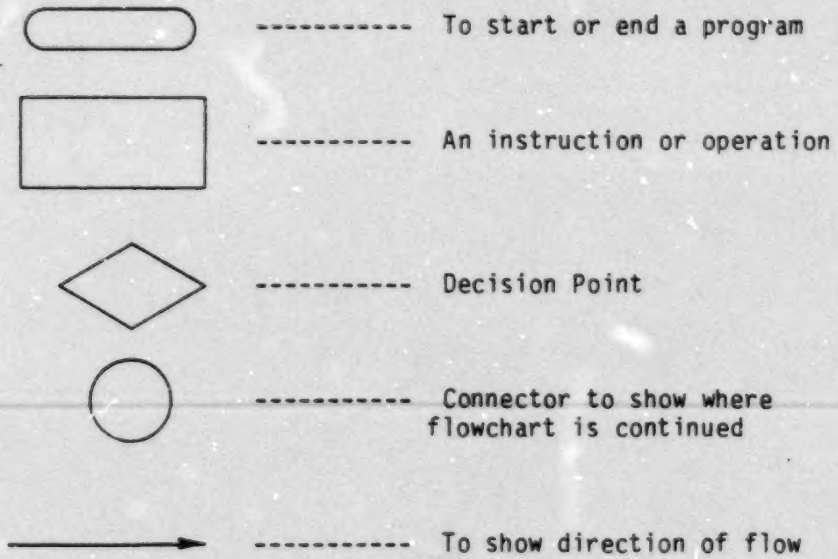


Once the objectives and general approach for videotex sequences have been determined, specific development begins. A critical element of this process is the flowchart, which becomes the roadmap for preparation of the content, script and computer programming. It serves as a central focus, not only to guide program development, but also to facilitate team interaction.

Exactly where in the design process a flowchart is prepared, for what purpose, and by whom depends on the complexity of the instructional sequence (in terms of branching requirements) and the expertise of team members. For simpler instructional sequences, the need for a flowchart becomes less important, and in some instances it may even be omitted. For example, testing sequences and branching structures where incorrect responses are immediately fed back into the main branch may not require a flowchart. In such cases, a content consultant may be able to prepare the basic script without the use of a flowchart by using some simple branching notation at decision points. One method is to simply list the page number where the student should be routed beside each choice. For testing sequences, the question and correct and incorrect feedback can be typed on one page. How the

branching is noted is really unimportant. What is important is that branch points be delineated so that someone unfamiliar with the content can correctly follow them. This procedure will insure that the script is correctly assembled and the computer programming correctly entered.

On the other hand, for preparation of complicated branching structures, especially those associated with gaming and simulation exercises, a flowchart becomes critical. A superior product can result when the flowchart is jointly developed with all team members having input. In this team setting, the instructional designer can provide input on appropriate instructional strategies, the content consultant can provide needed content expertise, the programmer can provide input about programming parameters, and the graphic designer can provide visualization suggestions. In preparing the flowchart, typical computer symbols can be used.


---------- To start or end a program
---------- An instruction or operation
---------- Decision Point
---------- Connector to show where flowchart is continued
---------- To show direction of flow

Although standard symbols have been developed for flowcharting, each project may have its own symbol requirements. For example, decision points are commonly used for "yes, no" options but are unwieldy when used with education's typical 4- or 5-choice multiple-choice format. Also, if a flowchart is being prepared as a guide to script development, it is important that enough content be presented so the purpose of each step is clear. Since a great deal of text usually needs to be presented, it is obvious that a diamond is a very inefficient symbol. In order to maximize text presentation, a rectangle may be a better choice to represent each screen -- whether it is an explanation frame or a decision point.

The first draft of the flowchart resulting from team meetings is likely to be a mess of erasures, crossouts and spaghetti-like paths. Team meetings are give-and-take sessions, characterized by false starts and both good and bad ideas. The first draft of the flowchart is generally redone in better form so that all team members can follow the logic.

In order to facilitate communication between team members, each symbol on the flowchart should be numbered. This number will provide easy identification and reference, and the number can also be used to identify script segments and completed frames. Because there are likely to be additions or because frames may become too long and have to be split, it is recommended that numbering proceed by tens. This procedure is standard in computer programming and allows the addition of frames/steps without renumbering.

In general, the more elements involved in a program, the more complicated the flowchart. It is important that the structure be

clear and communicate the intent to all team members, regardless of the form or the symbols used.

The flowchart should be scrutinized to be sure that no branches suddenly dead-end, that there are no closed branching loops from which the learner cannot escape, and that the branching structure allows the learner to proceed at his/her discretion with little confusion. It can serve as an excellent review tool, because it is the structural map of the program.

Figures 2 and 3 are presented to provide some examples of flowcharting.

Figure 2: Simple Interactive Strategy Showing Forward and Backward Branching

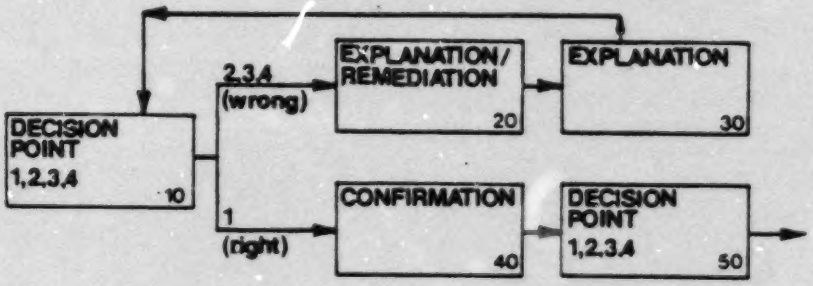
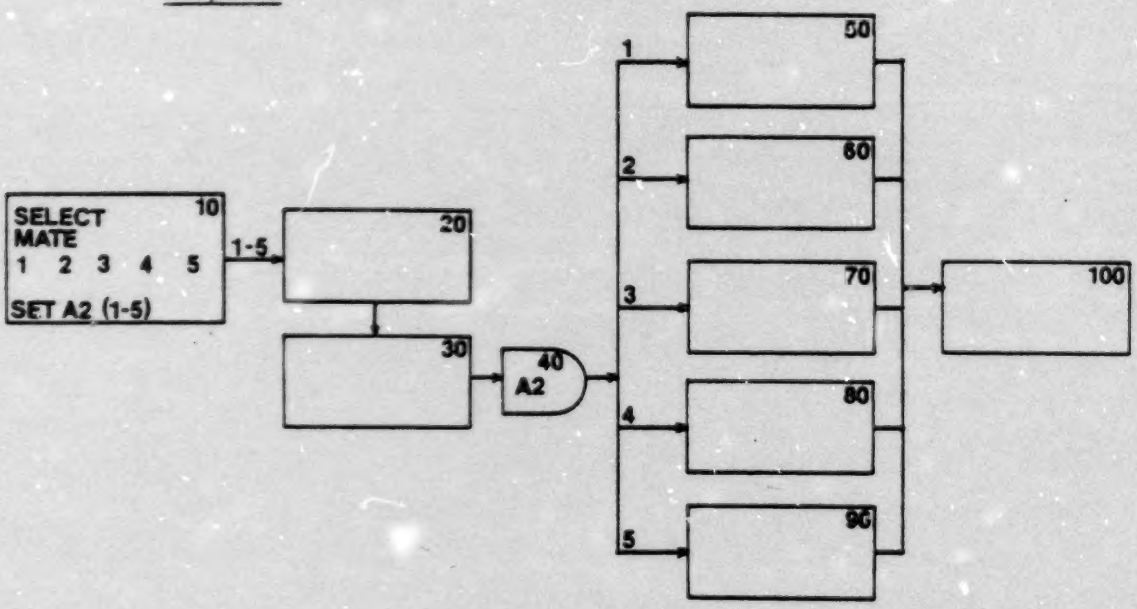


Figure 2 is a simple flowchart showing both forward and backward branching. Backward branching occurs when students make incorrect choices or want to try a different strategy. In such cases, they are backtracked to an earlier decision point and given the opportunity to select a different option.

More sophisticated computer programming requirements, such as computations (needed for grading) or recordkeeping, should be recorded on the flowchart to aid preparation of computer programming. If such operations have no bearing on script development, they can be omitted for that purpose. They should, however, be added before preparation of the computer programming. Different symbols can be used to graphically depict different computer operations. For example, branching based on the value of a single response or cumulative total of responses is shown in Figure 3.

Figure 3: Flowchart Showing Branching Based on Value of a Stored Response



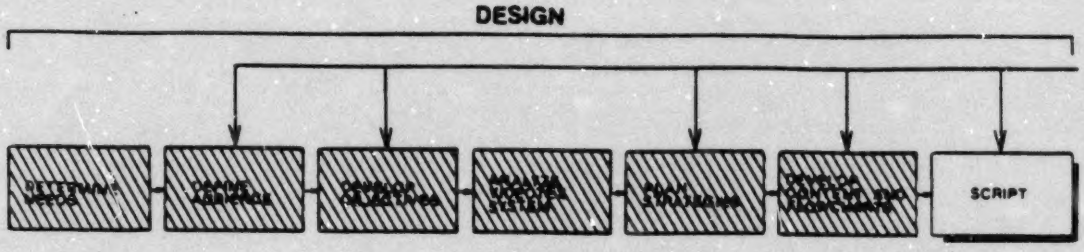
In the example above, the student makes a selection (1-5) at step 10 in the flowchart, and this response is recorded by the computer in storage point A2. At steps 20 and 30, the student is provided with additional information. At step 40, the student is branched to 50,

60, 70, 80 or 90, depending on the value recorded in A2. All branch paths then converge at step 100.

At step 40 no frame is displayed; it is a computer function only. Therefore, a different symbol has been used to distinguish it from steps where a frame is shown (rectangles). If there are steps where only computations are being performed, another symbol can be used. The importance of the symbols is that they provide a quick identification of what is happening at a particular step. In the example above, a reviewer could easily discriminate steps where frames are displayed and steps that involve only computer operations.

It should be reemphasized that the flowchart is used in different manners by the programmer and the scriptwriter. The writer is concerned with content and program flow in terms of the various paths and options open to the student. The programmer, on the other hand, could not care less about content. He or she is interested in program logic, the structure connecting the flowchart symbols, needed computations, etc. Since the flowchart must serve two purposes, it can become so overloaded with text and notations that the content explanations and computer statements compete for prominence and require "fishing out" by the respective user. One solution to the problem is preparing an original copy with only the symbols and directional flow and making two copies--one to contain the content information for the scriptwriter and the other with computer information for the programmer.

SCRIPTING



General Considerations

Scripts for videotex programs must account for the visual and text display, just as televison and film scripts delineate video and audio. As videotex instruction becomes more interactive, however, the resemblance to film/television and even slide/tape scripts decreases dramatically. A videotex instructional segment can consist of hundreds of frames whose relationship is only clear from examination of the flowchart. In working with such instruction it is critical to think nonlinearly. The writer may prepare one sequence, and depending on the complexity of the branching structure, prepare a follow-up sequence weeks later. The various tracks or story lines must be kept firmly in mind because the instruction is not likely to be written sequentially.

Another consideration is that videotex is not a replacement for a book. Studies have shown that users do not like dense frames (Carey, 1981), and it is usually recommended that frames contain only 50-70 words. Part of the difficulty is the limited number of words that can be presented on a screen. Most systems use a 32 x 16 or 40 x 20

format (number of columns x number of rows), which greatly reduces the amount of print that can be displayed at one time. Continuous pages with large amounts of text, especially if they do not solicit a response from the student, would be better handled in print form. Unfortunately, educational materials, and particularly those for higher-level audiences, tend to be very verbal and require considerable text presentation. To limit the amount of text display, information should be concise and tightly edited. This conciseness can become a strength. Students in videotex evaluations have noted that videotex material presents only pertinent information, while print information may present a great deal of extraneous information (Montgomerie, 1982, p. 56; Goldman, et al., 1982, p. 35).

Another consideration is the reading level of the material and its appropriateness for the audience. Special care should be taken in developing programs for younger children and special audiences, such as the hearing impaired. For materials oriented toward younger audiences, for example, the developer might consider spot checking the materials using some of the better known readability formulas such as the Maginnis Readability Graph (Maginnis, 1969) and Dale Chall Readability Formula (Dale & Chall, 1948).

To improve readability, sentences should be short. Semicolons should be avoided. It is better to use two sentences. Care should also be taken to avoid breaking paragraphs and particularly sentences onto subsequent frames. In cases where there is no alternative but to split a paragraph, the reader could be alerted to the continuation with a heading, e.g., "second paragraph continued."

Script Format

As discussed in the previous section, the flowchart becomes a guide to script development. The script expands upon the cryptic content notations in each flowchart symbol and contains information necessary for production of the individual frames. Cross referencing between the script and flowchart is accomplished by identifying each scripted frame (page number) with its corresponding reference number on the flowchart.

The three-column script form found in Figure 4 is the videotex script format used by the Nebraska project. This form was modified from one developed by Cox Cable for INDAX. A separate script page is devoted to each frame. The first column (far left) is used for descriptions of graphics. Any suggestions for visuals from the content consultant or instructional designer are noted here. When the graphic designer reviews the script, his or her ideas are also indicated in this column so that the final script reflects visual components of the completed frame. The middle column is for text presentation and reflects the INDAX 32 x 16 screen specifications. It allows for 16 lines of 32 characters each.

Line 1 is reserved for a frame and lesson identifier and/or identification number. A frame identification is often used to specify the topic of the lesson, such as "Memos" or "Adam Smith." An identification number is assigned by the development staff according to system specifications and project needs. To provide another cross-reference, the number can incorporate the one assigned at the flowchart stage. It can be used to aid in debugging (to be sure branches connect the correct frames) and to provide a means of identification and location for a student.

Figure 4: Sample Script Form

POLITICAL SCIENCE

SUBJECT: FASCISM SELF-CHECK #4

PAGE 640F___

FRAME #NPS01350640

NODE # 350640

Designer's Production Instructions and Specifications		Text: 32 characters by 16 lines double-spaced	elite pica		Copy-editing, Proofing
	1		.	1	
	2	FASCISM SELF-CHECK 350640	.	2	
	3		.	3	
RIGHT DONE AS DISPLAY	4	RIGHT	.	4	
TEXT IN LARGE LETTERS	5		.	5	
	6	EMPHASIS ON ACTION AS OPPOSED TO	.	6	
	7	THOUGHT OR CONSISTENCY IS	.	7	
	8	CHARACTERISTIC OF FASCISM.	.	8	
	9		.	9	
	10		.	10	
	11		.	11	
	12		.	12	
	13		.	13	
	14		.	14	
	15		.	15	
	16	PRESS NEXT FOR NEXT QUESTION	.	16	

BRANCHING INFORMATION:

SPECIAL INSTRUCTIONS:

Although it is usually recommended that some identification be used, keep in mind that lessons which are text-intensive often require a full 14 or 15 lines of print. In such cases the frame becomes even more dense with a print identifier on Line 1. To improve aesthetics and readability this identification can be omitted. To insure consistency, however, lessons should follow a similar format.

Line 16 (and sometimes line 15) is reserved for routing instructions to the learner. The following routing instructions have been used in the Nebraska Project:

1. PRESS NEXT - Used when the student is to advance to a subsequent page or frame
2. ENTER SELECTION - Used at a menu or decision frame when a student is presented with several numbered options and must input a choice
3. PRESS BACK TO REVIEW - Generally used in conjunction with another command (such as "OR PRESS NUMBER") when information relevant to a particular decision point must be presented on separate frames
4. PRESS MENU - Used when a student is to return to a major menu (such as a glossary menu) to make another selection

Because the computer will accept any numerical input, routing instructions are sometimes frame specific. For example, a frame requesting that a student enter his or her age displays the routing instruction "ENTER NUMBER OF YEARS."

The third column is used for copy-editing and proofing notes and questions. When working in a team setting, several people usually

must sign off on the script, and this column provides a space for corrections.

At the bottom of the form is a space for branching information. This area is used for any computer programming notes, such as a notation that a student response is to be recorded at this step. This space is also utilized to note branching information when no flowchart has been developed. In such cases, a simple notation can be used. "1-42, 2-43, 3-44, 4-60" would mean that if a student has selected choice 1, he or she should be branched to page 42, etc.

The "special instructions" area on the form is used for whatever needed. Often it is used by the instructional designer to note or explain some unusual feature of the frame to the person who will be inputting, e.g., "Representative is deliberately misspelled in this frame. Type it exactly as it appears in the script."

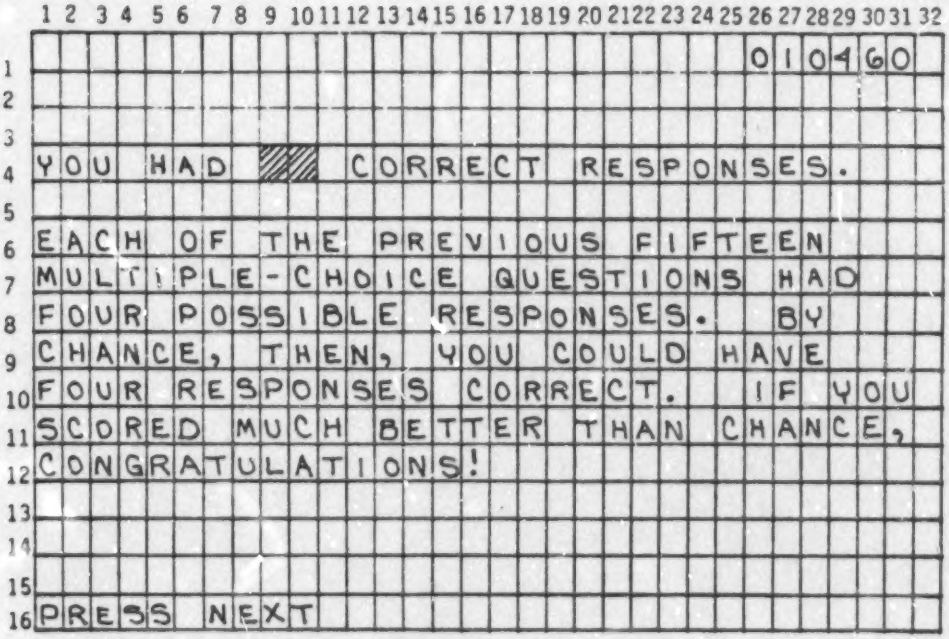
The script form used in Figure 4 is a general purpose form. In special cases, however, a more detailed form may be needed. Such a form is shown in Figure 5. This form represents a 16 x 32 grid. When text placement is critical and must be exact, this form is hand lettered showing precisely how the text should appear on the screen. One common use is when a blank must be left on the frame for display of a computer-generated grade or total. The computer program specifies the row and column where a grade should be displayed, and corresponding space must be left on the frame itself.

Figure 5: Script Format with Grid

PAGE 46OF___

FRAME #NSC01010460

NODE # 010460


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
010460
YOU HAD ### CORRECT RESPONSES.
EACH OF THE PREVIOUS FIFTEEN
MULTIPLE-CHOICE QUESTIONS HAD
FOUR POSSIBLE RESPONSES. BY
CHANCE, THEN, YOU COULD HAVE
FOUR RESPONSES CORRECT. IF YOU
SCORED MUCH BETTER THAN CHANCE,
CONGRATULATIONS!
PRESS NEXT

BRANCHING INFORMATION:

A3 DISPLAY AT 4,9-10

SPECIAL INSTRUCTIONS:

Summary

Careful scripting and flowcharting, although time consuming, can help assure the efficient development of interactive videotex segments. The flowchart, as the instructional roadmap, serves as an excellent communication tool for team members. Because it includes organization, structure, and programming elements, it serves as "common ground" for the various development functions. It is used to guide scripting, programming, and debugging.

The script fleshes out the flowchart and is used to guide frame development. Careful scripting assures material readability, conciseness, and logical division of content into ideas that can be communicated on separate frames. The script also provides specific instructions to the graphic artist or person who will be inputting the frame. The script, as the paper reflection of material to be communicated via the frames, is critical in the development process. With the flowchart, these two documents embody all instructional design, content, and production-related elements needed to move the instruction into the production stage.

This paper was extracted from Designing and Producing Videotex Instruction: A Producer's Handbook, prepared by the Nebraska Interactive Cable Project, 1983.

Bibliography

Amandam, K., & Kelly, J.T. Teaching and technology: closing the gap. T.H.E. Journal, November, 1982, 84-90.

Bork, A. Learning through graphics. In R. Seidel & M. Rubin (Eds), Computers and Communications: Implications for Education. New York: Academic Press, 1977.

Carey, John. Human factors in videotex. Unpublished manuscript prepared for the Corporation for Public Broadcasting, Office of Science and Technology, 1981.

CET In ɔrmation Sheet No. 2. Education and Information Technology. London: Council for Educational Technology, 1982.

Dale, E., & Chall, J.S. A formula for predicting readability: instructions. Educational Research Bulletin, 1948, 27, 37-54.

Dwyer, F. A guide for improving visualized instruction. State College, Pennsylvania: Learning Services, 1972.

Goldberg, F. Effects of imagery on learning incidental material in the classroom. Journal of Educational Psychology, 1974, 66, 233-237.

Goldman, R.J. Teletext transforms broadcast television into an interactive medium. CMLEA Journal, 1982, 5, 12-15.

Goldman, R.J., Gordon E., Craig, E., & Gingras, R.L. Teletext as an educational medium: the Los Angeles school trial. Los Angeles: KCET, 1982.

Goodfriend, K.K., Bamberger, N.J., Dozier, D.M., & Witherspoon, J.P. Final Report of the IPBS interactive videotex project. San Diego: KPBS, 1982.

Haring, M.J., & Fry, M.A. Effects of pictures on children's comprehension of written text. Educational Communication and Technology Journal, 1979, 27, 185-190.

Maginnis, G.D. The readability graph and informal reading inventories. The Reading Teacher, 1969, 22, 516-517.

Montgomerie, T.C. Telidon distance education field trial. Unpublished report commissioned by Alberta Education, Edmonton, Alberta, 1982.

OCLC. Channel 2000: description and findings of a viewdata test. Dublin, Ohio: Online Computer Library Center, 1980. (ERIC Document Reproduction Service No. 206-312)

Palmer, E.L. Formative research in the production of television for children. In D.R. Olson (Ed.), Media and symbols: the forms of expression, communication and education. Chicago: University of Chicago Press, 1974.

Peeck, J. Retention of pictorial and verbal content of a text with illustrations. Journal of Educational Psychology, 1974, 66, 880-888.

Public Broadcasting Report. Teletext field trial yields positive results but turns up some design and software problems. 1982, 4, 5.

Rasco, A.W., Tennyson, R.D., & Boutwell, R.C. Imagery instructions and drawings in learning prose. Journal of Educational Psychology, 1975, 67, 188-192.

Real, M. Videotex and education: a review of British developments. San Diego State University, 1981. (ERIC Document Reproduction Service No. 210-718)

Tennyson, R.D. Pictorial support and specific instructions as design variables for children's concept and rule learning. Educational Communications and Technology Journal, 1978, 26, 291-299.

Thompson, V. Using Prestel in educational applications. Proceedings of the 1982 International Videotex Conference. New York City, 1982, 113-121.

Trueman, M. Telidon and computer assisted learning - a report on the first experiment using Telidon for CAL. Proceedings of the 1982 International Videotex Conference. New York City, 1982, 123-131.

Tydeman, J., Lipinski, H., Adler, R., Nyhan, M., & Zwimpfer, L. Teletext and Videotex in the United States. New York: McGraw Hill, 1982.